Python Programmierung 1

23.08.2023 bis 01.11.2023

Durchführung: 23H__PYT1_22NI
Lehrperson: Walter Rothlin

Block	Seq	Dauer	Art	Inhalt	Ziele	Methode	Hausaufgabe
01	1	0.20L	PU	Vorstellung (Wer bin ich? Problem-Based Learning)			

01	2	1.20L	PU	**Installieren der Entwicklungsumgebung**
				* VNC / Notepad++ / PuTTY / FTP Client
				* Nano auf RPi, die wichtigsten Befehle
				* 1.Programm
				  * File erstellen und editieren (Console, Nano)
				  * print(), #! /usr/bin/pyhton3
				  * LINUX Befehle:
				  * cd, ls -al, pwd, rm, mv, cp, mkdir,
				  * LINUX file-system (chmod, filepath)
				  * Execution
				  * Fehlermeldungen interpretieren können und Lösungen implementieren
				* Notepad++, Putty
				* Programm erweitern (Print(), String-Operationen, Input())
					* RaspberryPi mit Sense-Hat in Betrieb nehmen und erstes «Hello.py» zur Ausführung bringen.
					* Ablaufstrukturen im Programm gezielt und richtig verwenden, print() und in-put() Funktionen sicher anwenden.
						Nach Anleitung installieren und konfigurieren	

01	3	2.60L	PU	**Aufgabe 1a (Umrechner.py)** * Menu * User-Input (Wähle:) * If-then-elif-else Struktur * Loop mit 0 beenden * Behandlung von falsch Eingaben * Formeln implementieren (Variablen, Float-Input, Math-Operatio-nen) * format() Methode	Umrechner.py (ohne eigene functions und ohne Bildschirmsteuerung)	Selber versuchen, Vormachen, Nachmachen mit theoretischen kurzen Einschüben	Umrechner.py alle Formeln implementieren und alle Menu-Punkte vollständig implementieren.
02	1	2.00L	PU	**Aufgabe 1b (Umrechner.py)** * Bildschirmsteuerung (cls(), halt()) implementieren * import math * Formeln in Funktionen implementieren * Funktionen abwärtskompatible erweitern * Exception Handling mit Pre-Checks und try-catch	Eigenen Funktion unter dem Aspekt der Re-Usability implementieren können.	Test-Driven Approach mit theoretischen Einschüben	
02	2	2.00L	PU	**Aufgabe 1c (Umrechner.py, xxLibrary.py)** * Funktion in eigene Library auslagern * Refactoring Umrechner.py verwendet eigene Library * Weitere Functionen readInt(), readFloat() implementieren, testen, anwenden und in eigene Library übernehmen. * Neuer Menu-Punkt: Quadratische Gleichung	Den Unterschied zwischen positional und named Parameters bei den Functions-Interfaces wie beim Aufruf sicher und gezielt anwenden können. Exception-Handling in Python sicher anwenden und den Unterschied zwischen pre-condition check and exception sich	Test-Driven Approach mit theoretischen Einschüben	Neue Funktionen entwickeln, testen und in eigene Lib übernehmen.
03	1	0.30L	PU	**Leistungskontrolle 1** * LINUX Basics	LINUX Basics	Moodle Test	

Block	Seq	Dauer	Art	Inhalt	Ziele	Methode	Hausaufgabe
03	2	1.20L	PU	**Aufgabe 2a (LED_Matrix.py)** 1. setPixel(), setPixels(), clear(), sleep(), showMessage() 2. Eventhandling (Joystick) 3. IMU- und Meteo-Sensoren	Alle Methoden im SenseHat Module (gemäss API doc) erfolgreich selbst getestet.	Test-Driven Approach mit theoretischen Einschüben	
03	3	2.80L	PU	**Aufgabe 2b (xx_SenseHat_Librarie.py)** 1. setPixel() mit clipping 2. drawLine(), drawRecantgle(), drawCircle() 3. Functions erweitern mit fillColor und borderColor 4. drawCompassNeedle(azimutInGrad)	Methoden Sense-Hat und Sense Klasse (API) mit LED Matrix verwenden.	Test-Driven Approach mit theoretischen Einschüben	Design und Implementation eines analogen Kompasses (mit Nadel)
04	1	1.80L	PU	Elemete in den verschiedenen Containers zugreifen (lesen), zufügen/ändern und löschen.	Containers in Python kennen und in eigenen Applikationen anwenden können.	Test-Driven Approach mit theoretischen kurzen Einschüben	
04	2	2.20L	PU	Sub-Listen mit \[1:-1\] ranges lesen resp verarbeiten/ändern. * for – Loops * Listen und Tuples * Dictonaries (keys())	Comprehensions mit Filter und ZIP für eigene Anwendungen einsetzen können.	Test-Driven Approach mit kurzen theoretischen Einschüben	Meteo-App oder einer Snake-App oder Linien Aufgaben
05	1	0.80L	PU	**Leistungskontrolle 2** Containers	Containers		
05	2	3.20L	PU	* Fakultät * Primzahlen Rechner * Primzahlen und Teiler Listen * Filter-Berechnungen	Algorithmen in Funktionen umsetzen Parameterüber-gaben * (listen) ** (dictonaries)	Test-Driven Approach mit kurzen theoretischen Einschüben	
06	1	0.80L	PU	Open-Weather REST Service mit eigenem Token (AppID) aus Python au-rufen (requesten) und response als JSON Struktur verarbeiten.	REST-Service mit JSON Response nutzen	Test-Driven Approach mit kurzen theoretischen Einschüben	
06	2	0.80L	PU	Filehandling open() for read, write and append (inkl UTF and ASCII)	Filehandling und direct EXCEL Zugriff erfolgreich anwenden	Test-Driven Approach mit kurzen theoretischen Einschüben	
06	3	2.20L	PU	Design und Implementation einer Meteo-Logger (Wetterstation), welche Meteo-Daten von einem Ort / Lokation optimiert und ohne «Löcher» loggen.	Filehandling und direct EXCEL Zugriff erfolgreich anwenden		Design und Implementation einer Meteo-Logger (Wetterstation), welche Meteo-Daten von einem Ort / Lokation optimiert und ohne «Löcher» loggen.
07	1	4.00L	PU	Eine eigene, allgemein einsetzbare Logger-Klasse gemäss Spezifikation entwickeln und testen. Anschliessend eigene Logger-Klasse in Meteo-App einsetzen.	Klassenkonzept in Python in einer konkreten Anwendung kennen lernen und anwenden können.	Test-Driven Approach mit kurzen theoretischen Einschüben	
08	1	0.20L	PU	Multi-Treathing und Timer-Events in Python kennen lernen.	Multi-Treathing und Timer-Events in Python kennen lernen.	Konzept anhand einiger Beispiele erklären (Walk-Through)	

Block	Seq	Dauer	Art	Inhalt	Ziele	Methode	Hausaufgabe
08	2	3.80L	PU	**Leistungsnachweis (Modullernzielkontrolle MILZ):** Eine allgemeine Weather-Class designen und implementieren, welche eine Wetterstation an einem bestimmten Ort kapselt.	Eine Wrapper-Class für einen Wetterdienst allgemein und nach OO Ansätzen designen und implementieren.	Selbstständiges programmieren und individuelle Reviews durch Dozenz.	
09	1	4.00L	PU	**Gebäudeautomation** * Selecta-Automat steuern * Rolladensteuerung anhand Wettervorhersagen * PiPlates * Shellys	REST Service komponieren aus bestehende-REST Services	Test-Driven Approach mit kurzen theoretischen Einschüben	